IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Appellant(s):	Bohannon et al.
Case:	13-13-1-1
Serial No.:	10/626,835
Filing Date:	July 24, 2003
Group:	2161
Examiner:	Paul Kim
Title:	Method and Apparatus for Composing XSL Transformations with XML Publishing Views

---

REPLY BRIEF

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Appellants hereby reply to the Examiner's Answer, mailed August 3, 2009 (referred to hereinafter as "the Examiner's Answer"), in an Appeal of the final rejection of claims 1-19 and 21-32 in the above-identified patent application.

REAL PARTY IN INTEREST

A statement identifying the real party in interest is contained in Appellants' Appeal Brief.

RELATED APPEALS AND INTERFERENCES

A statement identifying related appeals is contained in Appellants' Appeal Brief.

STATUS OF CLAIMS

A statement identifying the status of the claims is contained in Appellants' Appeal Brief.

STATUS OF AMENDMENTS

A statement identifying the status of the amendments is contained in Appellants' Appeal Brief.

SUMMARY OF CLAIMED SUBJECT MATTER

A Summary of the Invention is contained in Appellants' Appeal Brief.

STATEMENT OF GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A statement identifying the grounds of rejection to be reviewed on appeal is contained in Appellants' Appeal Brief.

CLAIMS APPEALED

A copy of the appealed claims is contained in an Appendix of Appellants' Appeal Brief.

ARGUMENT

Independent Claims

Independent claims 1, 10 and 24 were rejected under 35 U.S.C. §103(a) as being anticipated by Helgeson in view of Fernandez.

Helgeson

With regard to Helgeson, Appellants have argued, primarily, that Helgeson does not disclose or suggest "view queries." Column 80, lines 51-55, of Helgeson merely discloses that an XML document can be ereated from a database. While Helgeson may use the term "view" in the *presentation sense*, Helgeson does not use the term "view query," nor does Helgeson address using view queries to map between relational tables and a resulting XML document. As indicated in Appellants' prior responses, a "view query" specifies a *mapping between* the *relational tables* and the *resulting XML document*. When interpreting the term

"view query," the Examiner cannot ignore the word "query" and interpret the phrase "view query" to be, simply, a "view." If Appellants intended to claim a "view" they would have merely recited the word "view" in the claims. The Examiner's interpretation is at odds with the express claim language and the well-accepted definition of the term "view query."

In the Response to Arguments section, the Examiner again asserts that Appellants are arguing features not recited in the claims (referencing the mapping between tables and an XML document). To the contrary, however, the claims expressly recite a "view query" and Appellants are merely referencing the *well-accepted definition* of the term "view query" to those of ordinary skill in the art. The Examiner has again *not contested* this well-accepted definition of the term "view query" and is asserting an interpretation that is inconsistent with the express claim language ("view query," as claimed, versus "view" as suggested by the Examiner).

In addition, since Helgeson is not addressing "view queries," Helgeson does not disclose or suggest "***modifying the initial view query*** to account for an effect of said at least one transformation (specified in an XLST stylesheet)," or "applying said modified view query to said relational database to obtain said XML document," as further required by independent claims 1 and 24.

Fernandez

With regard to Fernandez, Appellants have argued, primarily, that Fernandez does not disclose or suggest "***modifying the initial view query*** to account for an effect of said at least one transformation (specified in an XLST stylesheet)," or "applying said modified view query to said relational database to obtain said XML document," as further required by independent claims 1 and 24.

To the extent that the XML view-definition query of Fernandez is a "view query," there is no disclosure or suggestion that this view query is ***modified*** at all, *or especially*, "to account for an effect of said at least one transformation." Fernandez merely teaches that an XML view can be specified by a query in a declarative query language of a middleware system. See, col. 3, lines 25-27. In the Response to Arguments section, the Examiner references col. 3, lines 11-22 of Fernandez for its alleged disclosure of mapping *relational sources to XML views*. Appellants can find no teaching in this passage, however, of modifying a "*view query*" (which, as already noted, requires *mapping between* the *relational* ***tables*** and the *resulting XML* ***document***).

The Examiner asserts that taking "a user query and the RXL (view) query and generat(ing) a new RXL query," suggests "modifying the initial view query *to account for an effect of said at least one transformation*." First, the transformation required by the claims must be specified in an XLST stylesheet. The Examiner has not established this point, as discussed further in the following paragraph. Second, the Examiner appears to have mischaracterized Fernandez. In fact, in the relevant discussion, at col. 3, lines 17-20, Fernandez states that it provides a "query composition algorithm that, when given an RXL query and an XML-QL query, generates a new RXL query equivalent to their composition." *This discussion is limited to composition of* ***"queries"*** and not "view queries," which, as already noted, require *mapping between* the *relational* ***tables*** and the *resulting XML* ***document***. In addition, the described *query* modification composes the user query and the RXL query to generate an equivalent composite query. There is no suggestion of *accounting for a transformation*.

The Examiner asserts that "wherein the user query ... is dependent on an XML template, which defines the XML document, any transformation to said XML document would result in the reflection of said transformation with the XML-QL user query." The Examiner has not identified any particular support for this statement, and Appellants cannot find a discussion in Fernandez that supports this passage. It appears to be a hypothetical scenario created by the Examiner. Even if Fernandez does disclose modifying an XML-QL user query in this manner, a "user query" is a query and not a "view query," as indicated above.

Thus, Fernandez does not disclose or suggest "***modifying the initial view query*** *to account for* an effect of said at least one transformation (specified in an XLST stylesheet)," or "applying said modified view query to said relational database to obtain said XML document," as further required by independent claims 1 and 24.

Functional Claim Language Must Be Considered

The Examiner asserts that the limitation "to obtain said XML document" should not be given patentable weight, as it is "an intended use." To the contrary, however, this positively recited limitation provides additional functional language to describe the present invention and must be given patentable weight. This limitation cannot be ignored. A functional limitation must be evaluated and considered, just like any other limitation of the claim, for what it fairly conveys to a person of ordinary skill in the pertinent art in the context in which it is used.

MPEP 2173.05(g). A functional limitation defines a particular capability or purpose that is served by the recited element, ingredient or step. *Id.*

KSR Considerations

Thus, even as combined in the manner suggested by the Examiner, Helgeson and Fernandez *do not teach every element of the independent claims*. Furthermore, based on the KSR considerations discussed hereinafter, the combination/modification suggested by the Examiner is not appropriate. Other than to allege that the motivation to combine is the desire to obtain the result ("in order to obtain an XML document according to the transformation specified by an XSLT stylesheet"), the Examiner has failed to establish "an apparent reason to combine … known elements." *KSR International Co. v. Teleflex Inc. (KSR)*, 550 U.S. ___, 82 USPQ2d 1385 (2007). Appellants query how this suggests an alleged combination to *modify the initial view query* to account for an effect of said at least one transformation (specified in an XLST stylesheet)," or "applying said modified view query to said relational database to obtain said XML document." As discussed below, this is insufficient to satisfy the Examiner's burden of proof under *KSR*.

Appellants are claiming a new technique for exporting at least a portion of a relational database to an XML document **by** "*modifying the initial view query* to account for an effect of said at least one transformation (specified in an XLST stylesheet)," and "applying said modified view query to said relational database **to obtain** said XML document."

Initial view queries are not merely recited as an element of each claim, but rather the functional language requires that the initial view queries be modified *to account for a transformation* and then applied to a relational database to obtain the result. There is no suggestion in Helgeson or in Fernandez, alone or in combination, to modify initial view queries to account for a transformation and then applied to a relational database to obtain the result. In addition, there is absolutely no suggestion to modify the ***views*** (purely in the presentation sense) of Helgeson with the ***initial view queries*** of Fernandez.

Claim 10

With regard to claim 10, the Examiner again asserts that Helgeson discloses composing an XSLT stylesheet (citing col. 51, lines 32-34) and Fernandez discloses (with) an XML view on said relational database to produce said modified view query (citing column 2, lines 65-67, and col. 3, lines 23-54). Appellants can find no disclosure or suggestion in

Fernandez of generating a modified view query. The passages referenced by the Examiner merely teach that an XML view can be specified by a query in a declarative query language of a middleware system. See, col. 3, lines 25-27. There is no disclosure or suggestion that this view query is **modified**. Claim 10 emphasizes that the modified view query is generated against a relational database to produce an XML document.

Appellants submit that the KSR considerations addressed above with regard to claims 1 and 24 are equally applicable to claim 10.

Appellants respectfully request the withdrawal of the rejection of independent claims 1, 10, and 24.

Dependent Claims

Claims 2-9, 11-18, 20-23 and 25-32 are dependent on independent claims 1, 10, 19 and 24, respectively, and are therefore patentably distinguished over each of the cited references, alone or in combination, because of their dependency from independent claims 1, 10, 19 and 24, for the reasons set forth above, as well as other elements these claims add in combination to their base claim.

Claims 4-5, 7, 12-14, 16, 21-22, 27-28 and 30 were indicated to be allowable if rewritten in independent form. **Claims 8, 17 and 23 appear to be rejected, but they depend on allowed claims 7, 16 and 22, respectively.** Thus, Claims 8, 17 and 23 are also believed to be allowable if rewritten in independent form.

Dependent claims 3, 11 and 26 require the steps of generating a first graph representing processing done by said XSLT stylesheet; and combining said first graph with a second graph representing said initial view query by matching pairs of nodes from the first and second graphs. Contrary to the Examiner's assertion, O'Carroll does not disclose or suggest an initial view query at all, nor a second graph representing said initial view query. A "view query" specifics a *mapping between* the *relational tables* and the *resulting XML document.* This is not shown by O'Carroll.

Dependent claims 6, 15 and 29 require the steps of pruning said combined graph to remove unnecessary nodes; and modifying said combined graph to produce a modified view query that handles formatting instructions. Contrary to the Examiner's assertion, Mani does not disclose or suggest a modified view query at all, nor a modified view query that handles

formatting instructions. A "view query" specifies a *mapping between* the *relational tables* and the *resulting XML document*. This is not shown by Mani.

Conclusion

All of the pending claims following entry of the amendments, i.e., claims 1-19 and 21-32, are in condition for allowance and such favorable action is earnestly solicited.

If any outstanding issues remain, or if the Examiner or the Appeal Board has any further suggestions for expediting allowance of this application, the Examiner and the Appeal Board are invited to contact the undersigned at the telephone number indicated below.

The attention of the Examiner and the Appeal Board to this matter is appreciated.

Respectfully submitted,

Kevin M. Mason

Date: October 5, 2009

Kevin M. Mason
Attorney for Appellants
Reg. No. 36,597
Ryan, Mason & Lewis, LLP
1300 Post Road, Suite 205
Fairfield, CT 06824
(203) 255-6560

APPENDIX

1. A method for exporting at least a portion of a relational database to an XML document, comprising the steps of:

obtaining an initial view query that defines an XML view on said relational database and an XSLT stylesheet specifying at least one transformation;

modifying said initial view query to account for an effect of said at least one transformation; and

applying said modified view query to said relational database to obtain said XML document.

2. (Original) The method of claim 1, wherein said XSLT stylesheet is based on a restrictive subset of XSLT.

3. The method of claim 1 further comprising the steps of generating a first graph representing processing done by said XSLT stylesheet; and combining said first graph with a second graph representing said initial view query by matching pairs of nodes from the first and second graphs.

4. The method of claim 3, wherein said combined graph is a context transition graph for an XSLT stylesheet executed on said initial view query.

5. The method of claim 4, wherein said context transition graph captures context transitions that occur when evaluating said XSLT stylesheet on said XML document produced by said initial view query.

6. The method of claim 3, further comprising the steps of pruning said combined graph to remove unnecessary nodes; and modifying said combined graph to produce a modified view query that handles formatting instructions.

7. The method of claim 6, further comprising the step of generating a traverse view query from a context transition graph prior to generating said modified view query, said traverse

view query capturing traversal actions of said XSLT stylesheet on said XML document produced by said initial view query.

8. The method of claim 7, wherein said formatting instructions are expressed as output tag trees for each node in said traverse view query; and further comprising the step of combining said output tag trees and said traverse view query to generate said modified view query.

9. The method of claim 1, wherein said obtained XML document is substantially similar to a second XML document produced by applying said XSLT stylesheet on said XML document produced by said initial view query.

10. A method for generating a modified view query against a relational database to produce an XML document, comprising the step of:

composing an XSLT stylesheet with an XML view on said relational database to produce said modified view query.

11. The method of claim 10, further comprising the steps of generating a first graph representing processing done by said XSLT stylesheet; and combining said first graph with a second graph representing an initial view query that defines said XML view on said relational database by matching pairs of nodes from the first and second graphs.

12. The method of claim 11, wherein said combined graph is a context transition graph for an XSLT stylesheet executed on said initial view query.

13. The method of claim 12, wherein said context transition graph captures context transitions that occur when evaluating said XSLT stylesheet on said XML document produced by said initial view query.

14. The method of claim 11, wherein said context transition graph includes selecting and matching transformations from said XSLT stylesheet.

15. The method of claim 11, further comprising the steps of pruning said combined graph to remove unnecessary nodes; and modifying said combined graph to produce a modified view query that handles formatting instructions.

16. The method of claim 15, further comprising the step of generating a traverse view query from a context transition graph prior to generating said modified view query, said traverse view query capturing traversal actions of said XSLT stylesheet on said XML document produced by said initial view query.

17. The method of claim 16, wherein said formatting instructions are expressed as output tag trees for each node in said traverse view query; and further comprising the step of combining said output tag trees and said traverse view query to generate said modified view query.

18. The method of claim 10, wherein an obtained XML document is substantially similar to a second XML document produced by applying said XSLT stylesheet on said XML document produced by said initial view query.

19. A method for generating a modified view query against a relational database to produce an XML document, comprising the steps of:

generating a first graph representing processing done by an XSLT stylesheet;

combining said first graph with a second graph to generate a combined graph representing a view query that defines an XML view on said relational database by matching pairs of nodes from the first and second graphs, wherein said combined graph is a context transition graph for said XSLT stylesheet executed on an initial view query that defines said XML view on said relational database;

pruning said combined graph to remove unnecessary nodes; and

modifying said combined graph to produce said modified view query that handles formatting instructions.

20. (Cancelled).

21. The method of claim 19, wherein said context transition graph captures context transitions that occur when evaluating said XSLT stylesheet on said XML document produced by said initial view query.

22. The method of claim 19, further comprising the step of generating a traverse view query from said context transition graph prior to generating said modified view query, said traverse view query capturing traversal actions of said XSLT stylesheet on said XML document produced by said initial view query.

23. The method of claim 22, wherein said formatting instructions are expressed as output tag trees for each node in said traverse view query; and further comprising the step of combining said output tag trees and said traverse view query to generate said modified view query.

24. A system for exporting at least a portion of a relational database to an XML document, comprising:

a memory; and

at least one processor, coupled to the memory, operative to:

obtain an initial view query that defines an XML view on said relational database and an XSLT stylesheet specifying at least one transformation;

modify said initial view query to account for an effect of said at least one transformation; and

apply said modified view query to said relational database to obtain said XML document.

25. The system of claim 24, wherein said XSLT stylesheet is based on a restrictive subset of XSLT.

26. The system of claim 24, wherein said processor is further operative to generate a first graph representing processing done by said XSLT stylesheet; and combine said first graph

with a second graph representing said initial view query by matching pairs of nodes from the first and second graphs.

27. The system of claim 26, wherein said combined graph is a context transition graph for an XSLT stylesheet executed on said initial view query.

28. The system of claim 27, wherein said context transition graph captures context transitions that occur when evaluating said XSLT stylesheet on said XML document produced by said initial view query.

29. The system of claim 26, wherein said processor is further operative to prune said combined graph to remove unnecessary nodes; and modify said combined graph to produce a modified view query that handles formatting instructions.

30. The system of claim 29, wherein said processor is further configured to generate a traverse view query from a context transition graph prior to generating said modified view query, said traverse view query capturing traversal actions of said XSLT stylesheet on said XML document produced by said initial view query.

31. The system of claim 30, wherein said formatting instructions are expressed as output tag trees for each node in said traverse view query; and further comprising the step of combining said output tag trees and said traverse view query to generate said modified view query.

32. The system of claim 24, wherein said obtained XML document is substantially similar to a second XML document produced by applying said XSLT stylesheet on said XML document produced by said initial view query.

EVIDENCE APPENDIX

There is no evidence submitted pursuant to § 1.130, 1.131, or 1.132 or entered by the Examiner and relied upon by appellant.

RELATED PROCEEDINGS APPENDIX

There are no known decisions rendered by a court or the Board in any proceeding identified pursuant to paragraph (c)(1)(ii) of 37 CFR 41.37.